13 November 2000
00-TFB-1204

Mr. Chris Wiltsee
NASA Ames Research Center
Mail Stop 244-30
Moffett Field, CA 94035-1000

Dear Mr. Wiltsee:

In accordance with the Document Requirements of Contract NAS2-97001, Stratospheric Observatory for Infrared Astronomy, Universities Space Research Association is pleased to provide PM20-007 Software Assurance Plan for the Mission Controls and Communications System (MCCS), a type 1 document.

If we need to discuss any of this material, please do not hesitate to call.

Respectfully,

Tom F. Bonner, Jr.
USRA SOFIA Project Manager

TFB/lf

SOFTWARE MANAGEMENT PLAN

PM 20-007
Revision -

SOFTWARE ASSURANCE PLAN
FOR THE
MISSION CONTROLS AND COMMUNICATIONS SYSTEM
(MCCS)

STRATOSPHERIC OBSERVATORY FOR INFRARED ASTRONOMY

(SOFIA)

Contract NAS2-97001

SOFTWARE MANAGEMENT PLAN

PM 20-007
Revision -

SOFTWARE ASSURANCE PLAN FOR THE MISSION CONTROLS AND COMMUNICATIONS SYSTEM (MCCS)

Prepared By:

Universities Space Research Association (USRA)
7500 Maehr Road
Waco, Texas 76705

Approved: /s/ Robert J. Hovde **Date:** 11/13/00

Robert J. Hovde
Software Development Management Team Leader

Approved: /s/ Tom F. Bonner **Date:** 11/14/00

Tom F. Bonner
Project Manager

DOCUMENT NUMBER: PD96100016-000

REVISION: -

23 October 2000

SOFTWARE ASSURANCE PLAN FOR THE MISSION CONTROLS & COMMUNICATIONS SYSTEM (MCCS) OF THE SOFIA PROJECT

SUBCONTRACT NUMBER: USRA-8500-02
CDRL ITEM: ROLLED OUT FROM PM20

PREPARED BY:

RAYTHEON COMPANY AIRCRAFT INTEGRATION SYSTEMS
P.O. Box 154580
Waco, Texas 76715-4580

PREPARED FOR:

UNIVERSITIES SPACE RESEARCH ASSOCIATION
7500 Maehr Road
Waco, Texas 76705

PREPARED BY:	/s/ Barry Papke	DATE:	11 Nov 00
	Barry Papke, MCS IPT Lead		
REVIEWED BY:	/s/ Nicholas A. Delp	DATE:	13 Nov 2000
	Nicholas Delp, Configuration Manager		
	/s/ Sheldon D. Jordan	DATE:	13 Nov 2000
	Sheldon D. Jordan, Software Quality Assurance		
	/s/ J. Gregory Bruich	DATE:	11 Nov 2000
	J. Gregory Bruich, MCCS Technical Director		
	/s/ Gregory Arceneaux	DATE:	13 Nov 2000
	Gregory Arceneaux, SEIT IPT Lead		
	/s/ Arthur L. Scheuermann	DATE:	13 Nov 2000
	Arthur Scheuermann, Technical Director		
APPROVED BY:	/s/ Dennis Boroczk	DATE:	13 Nov 2000
	Dennis Boroczk, Project Manager		

REVISIONS

Revisions to the document from the previous issue are denoted by vertical bars in the margin of the page.

REV	DATE	DESCRIPTION	APPROVAL

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

1.1 IDENTIFICATION OF DOCUMENT

This document is the Software Assurance Plan for the Mission Controls and Communication System (MCCS) of the Stratospheric Observatory for Infrared Astronomy (SOFIA) Project as required for SOF-1017 Data Requirement PM20. It is the roll-out document for the software assurance section of PM20-008, MCCS Software Management Plan for the SOFIA Project.

1.2 SCOPE OF DOCUMENT

This document applies to the software for which Raytheon Company Aircraft Integration Systems (RAIS) is responsible, the MCCS software consisting of the Mission Control Subsystem (MCS). It applies to the development phase (CLIN 1) of the SOFIA project only.

1.3 PURPOSE AND OBJECTIVES OF DOCUMENT

The purpose of this document is to specify the assurance process to be followed in developing the software for SOFIA. It encompasses the responsibilities, standards, procedures, and organizational relationships necessary to carry out this process. The process consists of these activities:

- Quality assurance
- Verification and validation (V&V)
- Quality engineering assurance (Note: In some MCCS documents these requirements are referred to as “specialty engineering” or “-ilities”)
- Safety assurance
- Security and privacy assurance
- Certification

The general program view of the assurance process is given in the Software Quality Assurance Plan PM21-003 section 5.1.3.

1.4 DOCUMENT STATUS AND SCHEDULE

This is the initial release of the Software Assurance Plan for the MCCS.

1.5 DOCUMENT ORGANIZATION

No specific format is required by SOF-1017 DR PM20. RAIS and its subcontractors have constructed this volume using the format defined for Data Item Description M400 of NASA Software Documentation Standard Software Engineering Program (NASA-STD-2100-91). There are no roll-outs of this document; however, some sections of this document refer to text in the parent document or in documents elsewhere in the SOFIA documentation tree.

Section 1 contains an introduction and overview of this document.

Section 2 lists related documentation.

Section 3 describes the software quality assurance plan. It references PM21-003, the Software Quality Assurance Plan.

Section 4 describes the verification and validation plan.

Section 5 describes the quality engineering assurance plan.

Section 6 describes the safety assurance plan. It references PM21-004, the Hazard Analysis.

Section 7 describes the security and privacy assurance plan.

Section 8 describes the certification plan. It references PM21-003, the Software Quality Assurance Plan.

Section 9 contains a list of acronyms and abbreviations.

Section 10, Glossary, provides an alphabetical listing of definitions of terms used for the SOFIA Project software.

Section 11, Notes, points to other sources of assurance information.

2.0 RELATED DOCUMENTATION

2.1 PARENT DOCUMENTS

PM20-008	MCCS Software Management Plan (supplement to PM20-001 for MCCS)

2.2 APPLICABLE DOCUMENTS

PM20-001	Software Management Plan for the SOFIA Project
PM20-002	Software Development Activities Plan for the MCCS (supplement to PM20-001)
PM20-005	MCCS Software Configuration Management Plan (supplement to PM20-001)
PM21-001	SRM&QA Plan for the SOFIA Project
PM21-002	Hardware Quality Assurance Plan for the SOFIA Project (supplement to PM21-001)
PM21-003	Software Quality Assurance Plan for the SOFIA Project (supplement to PM21-001)
PM21-004	Hazard Analysis for the SOFIA Project (supplement to PM21-001)
SOF-1017	SOFIA Data Requirements Specification
ANSI/ASQC Q-9001	Quality Systems-Model for Quality Assurance
ANSI/ASQC Q-9000-3	Guidelines for application of ISO 9001 to Software
AHB 5333.1	Establishment of Software Assurance Programs

2.3 INFORMATION DOCUMENTS

NASA-GB-A201	Software Assurance Guidebook
ANSI/IEEE Std 729-1983	IEEE Standard Glossary of Software Engineering Terminology

3.0 QUALITY ASSURANCE PLANNING

Software Quality Assurance (SQA) is the planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures. Software quality assurance planning for the MCCS software development lifecycle from system analysis through system integration and acceptance testing is specified in PM21-003, Software Quality Assurance Plan for the SOFIA Project.

3.1 APPROACH AND ACTIVITIES

In general, Software Quality Engineering (SQE) evaluates software management, engineering, configuration management, and subcontractor practices to assure compliance with the contract, the software program plans, and software-related procedures. SQE audits the software development process and its artifacts; evaluates requirements traceability from higher-level specifications to the design and code and through the test plans and procedures; reviews and approves deliverable software artifacts; participates in software-related formal reviews; verifies performance of peer reviews; monitors the software integration and build process; monitors metrics collection, analysis, and reporting; monitors software problem reporting and tracking; and participates (as a witness) in integration and acceptance testing where developmental software is a component of the system being tested.

3.2 METHODS AND TECHNIQUES

3.2.1 PROCESS ASSURANCE

Process audits will be performed by SQE on a monthly schedule primarily by means of Raytheon's AIS Process Definition (APD) software tool. All process audit findings (whether resulting from APD or non-APD audits) that are determined to be sufficiently serious to require formal tracking of corrective action to closure will be documented as Action Requests (ARs) using Raytheon's Action Request Data Base (ARDB) software tool. ARs effectively serve as CARs and the ARDB software tool serves as the in-process corrective action system for the software project. Audit results are also reported to functional management through SQE reports, and all SQA-related activities are reported in bi-weekly activity reports.

RAIS' management review of SQE and Software Engineering is conducted monthly during the Software Engineering Director's Software Program Review. Every monthly management review includes presentations, by the Software Process & Tools Manager and the Waco Products Manager, of various software metrics reports extracted from ARDB.

3.2.2 PRODUCT ASSURANCE

Product assurance is implemented at two levels of reviews. At the first level are the iterative work-in-progress document reviews. Artifacts that are reviewed at this level include software program planning documents (DR PM20); software engineering documents (DR SW); and software-related trade studies, white papers, and other technical reports. In the case of software DRs in particular, SQE participates fully throughout the review process, including final sign-off before release. The nature of this document review process precludes the generation of action requests or problem reports.

At the second level of the product assurance process are the several kinds of informal and semi-formal reviews of software development engineering artifacts. These artifacts (which make up the "technical" content of the SW DRs) include requirements, architectural and detailed designs, code, and integration and acceptance test procedures. Desk checks, walkthroughs, and code inspections are the principal kinds of informal reviews used by the software development team. Although SQE participates only minimally in informal reviews, those reviews do produce problem reports that are documented and tracked to closure using GNATS, the GNU Problem Report Management System. Semi-formal reviews, or peer reviews, are performed by the software development team with SQE's full participation using Raytheon's Peer Review Evaluation [software] Tool (PRET). All "accepted" peer review comments are documented and tracked to closure using GNATS. Besides participating fully in the peer review process, SQE monitors on a continuous basis the software development project's use of GNATS and reviews the monthly GNATS problem report (PR) status reports produced by SCM.

4.0 VERIFICATION AND VALIDATION PLANNING

4.1 APPROACH AND ACTIVITIES

4.1.1 RESPONSIBILITY

RAIS and its subcontractors will verify and validate the software they produce for SOFIA. The software verification and validation manager (SVVM) will manage all V&V activities, with oversight by the RAIS software quality assurance manager (SQAM). To the degree that the SQAM is under a separate management arm of RAIS and is an active participant in all V&V review and testing activities, there is independent V&V supervision throughout the life cycle.

Formal reviews mandated by NASA contract NAS2-97001 are not discussed in this document (even though they are V&V activities). The MCCS Software IPT conducts the formal reviews; these reviews are described in PM20-002, the MCCS Software Development Plan.

4.1.2 LIST OF ACTIVITIES

Verification and validation are closely tied to the development activities and products. The V&V plan therefore follows the organization of the software development life cycle, as described in the MCCS Software Development Activities Plan (PM20-002). Table 4.1.2-1 through Table 4.1.2-4 shows the list of V&V activities that RAIS and its subcontractors will perform and the correspondence between development activities and V&V activities. The sequence of activities in the tables is not indicative of any actual sequence of development activities for any software item, though the activities map directly to the development life cycle phases given in PM20-002 section 3.1.1. The actual sequence of activities will be defined in the planning for a given CSCI and incorporated into its Build/Iteration Plan. Also, the transition or completion criteria for each activity are specified in the procedures used for those activities.

Table 4.1.2-1. Table of Context Development Activities and V&V Activities

Development activity	Product	V&V activity / procedure	V&V product
Requirements Analysis and Specification	Product Specification (SW01)	Requirements review / PRET procedures	PRET review report and GNATS problem reports

Table 4.1.2-2. Table of Iterative Design Activities and V&V Activities

Development activity	Product	V&V activity / procedure	V&V product
Object oriented analysis	Class Responsibility and Collaboration (CRC) Review	CRC Cards	GNATS problem reports
Object oriented design	Design (Rose) Model and sequence diagrams	Architectural design review / PRET procedures	PRET review report and GNATS problem reports

Table 4.1.2-3. Table of Iterative Development Activities and V&V Activities

Development activity	Product	V&V activity / procedure	V&V product
Product iteration	Code	Desk check / Desk check procedure Code inspection (selected products only; inspect as resources permit) / PRET procedures	GNATS problem reports PRET review report
Product testing	Test execution / test procedures	Unit testing (as resources permit)	GNATS problem reports

Table 4.1.2-4. Table of Integration Activities and V&V Activities

Development activity	Product	V&V activity / procedure	V&V product
Integration testing	Test execution / test procedures	Integration testing (as resources permit)	GNATS problem reports Test Report

MCS verification and validation—like all V&V—is a resource-limited activity. The choice of how resources are allocated is based on the following criteria:

- All contractually deliverable data requirements (i.e., SW01-SW06) must be peer-reviewed at least once.
- All deliverable code will be verified by inspection (either Fagan inspection or desk check).
- All deliverable code will be verified by execution (in a unit or integration test).
- All user interfaces will be reviewed for human-factors compliance.

- Products that are the result of one or two individuals' work will be peer-reviewed more thoroughly than those that are the result of a collaborative effort.
- V&V efforts will concentrate on modules in which many faults have been found in earlier reviews and testing. (This criterion relies on the widely published observation that "bugs tend to cluster".)

Following each major build there is a period during which V&V activities are used to identify open issues in the product. Each CSCI will have its own Build and V&V activities schedule. Open issues and recording of test failures will follow the same problem reporting and change control processes used throughout the life cycle and defined in the MCCS Software Configuration Management Plan (PM20-005) section 4.2.

4.2 METHODS AND TECHNIQUES

4.2.1 GENERAL CONDUCT OF REVIEWS AND INSPECTIONS

Two forms of reviews are planned: formal reviews involving the entire MCS software IPT, with visibility by NASA, USRA, and the MCS development team, and reviews by the development team only. This document discusses only reviews by the development team. The general conduct of all these reviews will be as follows:

- The Software Verification and Validation Manager (SVVM) designates a recorder, moderator, and two or more inspectors for the item. If the item is the product of many developers, the MCS software IPT lead selects an author who has the responsibility of responding to the defects the review finds. This may involve the creation of a session using the RAIS Peer Review and Evaluation Tool (PRET) for the review.
- The Software Verification and Validation Manager (SVVM) or his designee circulates the review materials to every review participant at least 3 days prior to the meeting.
- The Software Configuration Management Manager (SCMM) initiates the process for collecting comments for this particular review (this may use the Raytheon PRET tool).
- Each inspector submits a list of proposed defects. The author is encouraged (but not required) to submit such a list as well.
- When the review is held, defects are accepted or rejected. The SCMM or a representative records these decisions and any supporting notes. These form the content of the review report, which is circulated to the team.
- The SVVM may convene a follow-up review or re-inspection to determine whether defects have been adequately addressed.

This review process is followed by all the review activities listed below with the exception of desk checks. In each case, the review materials package will take the form indicated.

4.2.2 REQUIREMENTS REVIEWS

The guidelines for the reviews will include criteria for completeness, necessity, consistency, clarity, logic, testability, and traceability. The reference documents for requirements reviews are the MCCS B-level and C-level specifications and the SOFIA contract NAS2-97001.

4.2.2.1 Objective

The objective of requirements reviews are to ensure the for completeness, necessity, consistency, clarity, logic, testability, and traceability of the allocated baseline.

4.2.2.2 Frequency and/or entry criteria

Requirements review meetings are performed incrementally as logical groups or functions are analyzed and the derived requirements are specified.

4.2.2.3 Preparations

Requirements white papers or other source documents are prepared and distributed for IPT review as defined in PM20-002. Following IPT review, the updated requirements list is prepared for internal peer review. Peer review preparation is as defined by the peer review procedures referenced in PM20-002.

4.2.2.4 Procedures reference

Peer Reviews

Peer Review and Evaluation Tool Manual

Procedures are available at http://sofia-usra.arc.nasa.gov/internal/mcs/mcs_ipt/.

4.2.2.5 Exit criteria

The review is complete when all Problem reports generated during the review are closed.

4.2.2.6 Products

Peer Review Report

Problem Reports

4.2.2.7 Reporting method

SW05 Inspection reports – (reference NASA-DID-R003) with PRET reports attached.

SW05 Review reports – (reference NASA-DID-R010) with comments from IPT reviews.

4.2.3 DESIGN REVIEWS

The guidelines for the reviews will include both design criteria and traceability to the requirements. The reference document for design reviews is the Software Requirements Specification (SW01).

4.2.3.1 Objective

The objective of design reviews is to ensure that the design is in compliance with all standards and guidelines, meets its allocated requirements, and has sufficient clarity and completeness that it can transition either on to detailed design or on to development.

4.2.3.2 Frequency and/or entry criteria

Design reviews are performed incrementally as the design of functions, classes or other logical groupings are completed. In addition, the entire system design is reviewed periodically at intervals of approximately 6 months. These reviews normally correspond to mid and final design reviews for each major build.

4.2.3.3 Preparations

The documentation of designs is as defined in PM20-002. The preparation for the review is as defined by the peer review procedure.

4.2.3.4 Procedures reference

Peer Reviews

Peer Review and Evaluation Tool Manual

Guidelines for Design Reviews

Procedures are available at http://sofia-usra.arc.nasa.gov/internal/mcs/mcs_ipt/.

4.2.3.5 Exit criteria

The review is complete when all Problem reports generated during the review are closed.

4.2.3.6 Products

Peer Review Report

Problem Reports

4.2.3.7 Reporting method

SW05 Inspection reports – (reference NASA-DID-R003) with PRET reports attached.

SW05 Review reports – (reference NASA-DID-R010) with comments from IPT reviews.

4.2.4 DESK CHECKS

Desk checks are code inspections for which only one inspector has been assigned. The reference documents for code are reviewed design documents.

4.2.4.1 Objective

The purpose is finding non-conformances, i.e. attributes of the inspected item that do not meet the established standards and guidelines.

4.2.4.2 Frequency and/or entry criteria

Code inspections occur after the design element has complete formal design review and the coded implementation of that component is deemed mature enough by the developer.

4.2.4.3 Preparations

The preparation for the review is as defined by the desk check procedure.

4.2.4.4 Procedures reference

Guidelines for Desk Checking Code

Procedures are available at http://sofia-usra.arc.nasa.gov/internal/mcs/mcs_ipt/.

4.2.4.5 Exit criteria

The review is complete when all Problem reports generated during the review are closed.

4.2.4.6 Products

Problem Reports

4.2.4.7 Reporting method

The state of all the code elements in the repository is changed from "beta" to "ready".

4.2.5 CODE INSPECTIONS

Formal code inspections will be conducted on selected software configuration items.

4.2.5.1 Objective

The purpose is finding non-conformances, i.e. attributes of the inspected item that do not meet the established standards and guidelines.

4.2.5.2 Frequency and/or entry criteria

Code inspections occur after the design element has complete formal design review and the coded implementation of that component is deemed mature enough by the developer.

4.2.5.3 Preparations

The preparation for the review is as defined by the peer review procedure.

4.2.5.4 Procedures reference

Peer Reviews

Peer Review and Evaluation Tool Manual

Guidelines for Code Inspections

Procedures are available at http://sofia-usra.arc.nasa.gov/internal/mcs/mcs_ipt/.

4.2.5.5 Exit criteria

The review is complete when all Problem reports generated during the review are closed.

4.2.5.6 Products

Peer Review Report

Problem Reports

4.2.5.7 Reporting method

SW05 Inspection reports – (reference NASA-DID-R003) with PRET reports attached.

SW05 Review reports – (reference NASA-DID-R010) with comments from IPT reviews.

The state of all the code elements in the repository is changed from "beta" to "ready".

4.2.6 UNIT TESTING

Unit tests may cover just some methods of a class, an entire class, or a class and its collaborators. The software testers will have full access to the design and code in order to perform white-box testing (i.e., testing that is planned by looking at the internal structure of the code). The test harness, test data, and test results will be archived.

In addition to observing the behavior of the unit under test through a test harness or stub, the testers will use memory leak detection, bounds checking, coverage, and timing tools to assess the performance of the code.

For a few modules, automated regression tests will be developed. Modules that show a high density of defects will be selected for automated regression testing. Modules whose unit test harnesses are easy to automate may also become part of the automated regression test suite.

To improve testing efficiency, software will be tested by different personnel than its developers whenever possible.

4.2.6.1 Objective

The purpose of unit testing is the early identification of coding errors or other implementation problems.

4.2.6.2 Frequency and/or entry criteria

As deemed necessary by the developer.

4.2.6.3 Preparations

N/A

4.2.6.4 Procedures reference

N/A

4.2.6.5 Exit criteria

The testing is complete when all Problem reports generated during testing are closed.

4.2.6.6 Products

Problem Reports

4.2.6.7 Reporting method

Problem Reports

SW05 Test reports - (reference NASA-DID-R009)

4.2.7 INTEGRATION TESTING

Integration testing uses the defined interface to the modules under test but does not generally delve into the internal structure (i.e., the planned interface tests are “black-box” tests). These tests will concentrate on the following areas:

- Adherence to external and internal interface control documents;
- Timing and resource (CPU, memory, disk, network, screen area, process table, object request, etc.) utilization;
- Behavior of software distributed amongst many threads, processes, and platforms;
- Investigation of behavioral problems found in usability testing.

As a goal, RAIS and its subcontractors will attempt to achieve 100% statement coverage through the combination of unit and integration tests.

4.2.7.1 Objective

The purpose of integration testing is to identify implementation problems across software components and between software item interfaces.

4.2.7.2 Frequency and/or entry criteria

Integration testing is performed at the end of each Build.

4.2.7.3 Preparations

Preparations are in accordance with informal integration plans.

4.2.7.4 Procedures reference

Performed in accordance with informal integration plans.

4.2.7.5 Exit criteria

The testing is complete when all Problem reports generated during testing are closed.

4.2.7.6 Products

Problem Reports

4.2.7.7 Reporting method

Problem Report metrics

SW05 Test reports - (reference NASA-DID-R009)

5.0 QUALITY ENGINEERING ASSURANCE PLANNING

Software quality engineering encompasses all activities that assure the characteristics of software *other than* its functionality. The MCCS has quality engineering requirements in the following areas:

- Performance and quality engineering (response time for commands, capacity for data input, and time synchronization)
- Human performance and engineering

5.1 APPROACH AND ACTIVITIES

5.1.1 PERFORMANCE AND TIMING

A statistical process will verify the timing and performance in testing performed at the end of each Build. In these tests, the tester or harness will initiate a large number of the transactions having the required behavior. The distribution of times and capacities is recorded. The tests can assess the likelihood that a given transaction will exceed the allotted time, or that a given capacity will not be available. When tests do not meet the performance requirements their results are recorded in problem reports for tracking to resolution.

5.1.2 HUMAN FACTORS

Human-factors requirements can be verified by the design reviews defined in section 4 above for those components with direct user interfaces.

6.0 SAFETY ASSURANCE PLANNING

These are no safety requirements allocated to software in the MCCS. Safety assurance is discussed in the Hazard Analysis, PM21-004.

7.0 SECURITY AND PRIVACY ASSURANCE PLANNING

There are no security or privacy requirements allocated to software in the MCCS so no software planning document will be generated..

8.0 CERTIFICATION PLANNING

There are no certification requirements for MCCS software separate from the AS/Observatory certification requirements so no software planning document will be generated.

9.0 ABBREVIATIONS AND ACRONYMS

APD	Aircraft Integration Systems (RAIS) Process Definition
AR	Action Request
ARDB	Action Request Database
AS	Aircraft System
CAR	Corrective Action Request
CDCS	Cavity Door Control System
CECS	Cavity Environment Control System
CLIN	Contract line item number
CSCI	Computer Software Configuration Item
FITS	Flexible Image Transport System
FP	Flight Planner
GNATS	the GNU problem report tracking system
IPT	Integrated Product Team
MCCS	Missions Controls and Communications Subsystem
MCS	Mission Control Subsystem
NASA-ARC	NASA Ames Research Center
OOSE	Object Oriented Software Engineering
PRET	Peer Review Evaluation Tool for RAIS
RAIS	Raytheon Aircraft Integration Systems
SCMM	Software Configuration Management Manager
SFH	Successful flight hours
SOFIA	Stratospheric Observatory for Infrared Astronomy
SQA	Software Quality Assurance
SQAM	Software Quality Assurance Manager
SRM&QA	Sustainability, reliability, maintainability, and quality assurance
SSMOC	SOFIA Science and Mission Operation Center
USRA	Universities Space Research Association
V&V	Verification and Validation
VCRI	Verification cross-reference index

10.0 GLOSSARY

The following definitions, taken from ANSI/IEEE Std 729-1983, are provided to clarify terms used in this document.

Formal testing: The process of conducting testing activities and reporting results in accordance with an approved test plan.

Qualification testing: **Formal testing**, usually conducted by the developer for the customer, to demonstrate that the software meets it specified requirements.

Testing: The process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements or to identify differences between expected and actual results.

Validation: The process of evaluating software at the end of the software development process to ensure compliance with software requirements.

Verification:

- The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase.
- Formal proof of program correctness.
- The act of reviewing, inspecting, **testing**, checking, auditing, or otherwise establishing and documenting whether or not items, processes, services, or documents conform to specified requirements.

11.0 NOTES

11.1 WHERE TO FIND OTHER ASSURANCE INFORMATION

The following sections of other documents in the SOFIA tree have at least some overlap with this document.

11.1.1 PM20-001A SOFTWARE MANAGEMENT PLAN FOR SOFIA

Section 6.3, “Statement of Applicable Standards”, briefly refers to ISO 9000 and the NASA Software Assurance and Documentation standards.

Section 8.5, “Security and Privacy Assurance Planning”, describes in details the processes used to ensure security of SOFIA software.

Section 13, “Glossary”, contains many definitions related to SQA that are not the same as the IEEE standard.

11.1.2 PM20-002A SOFTWARE DEVELOPMENT ACTIVITIES PLAN FOR THE MCCS

Section 3.1.3, “Applicable standards”, lists the standards and procedures on the Web site.

Section 3.1.6, “Metrics to be collected and assessed” includes development completion "DM" and "Fault profiles metric”.

Section 5.0, “Formal reviews” includes a list of all reviews.

Section 5.2, “Conduct of formal reviews”, describes in detail what activities occur during a formal review.

Section 5.3, “Software development reviews”, lists the informal reviews conducted during the software process.

11.1.3 PM20-005 SOFTWARE CONFIGURATION MANAGEMENT PLAN FOR SOFIA

Section 4.2, “Change Control”, defines the change control process.

11.1.4 PM21-003 SOFTWARE QUALITY ASSURANCE PLAN FOR THE MCCS

Section 5.1 includes Table 5.1.2-1 Software Quality Lifecycle Activities.